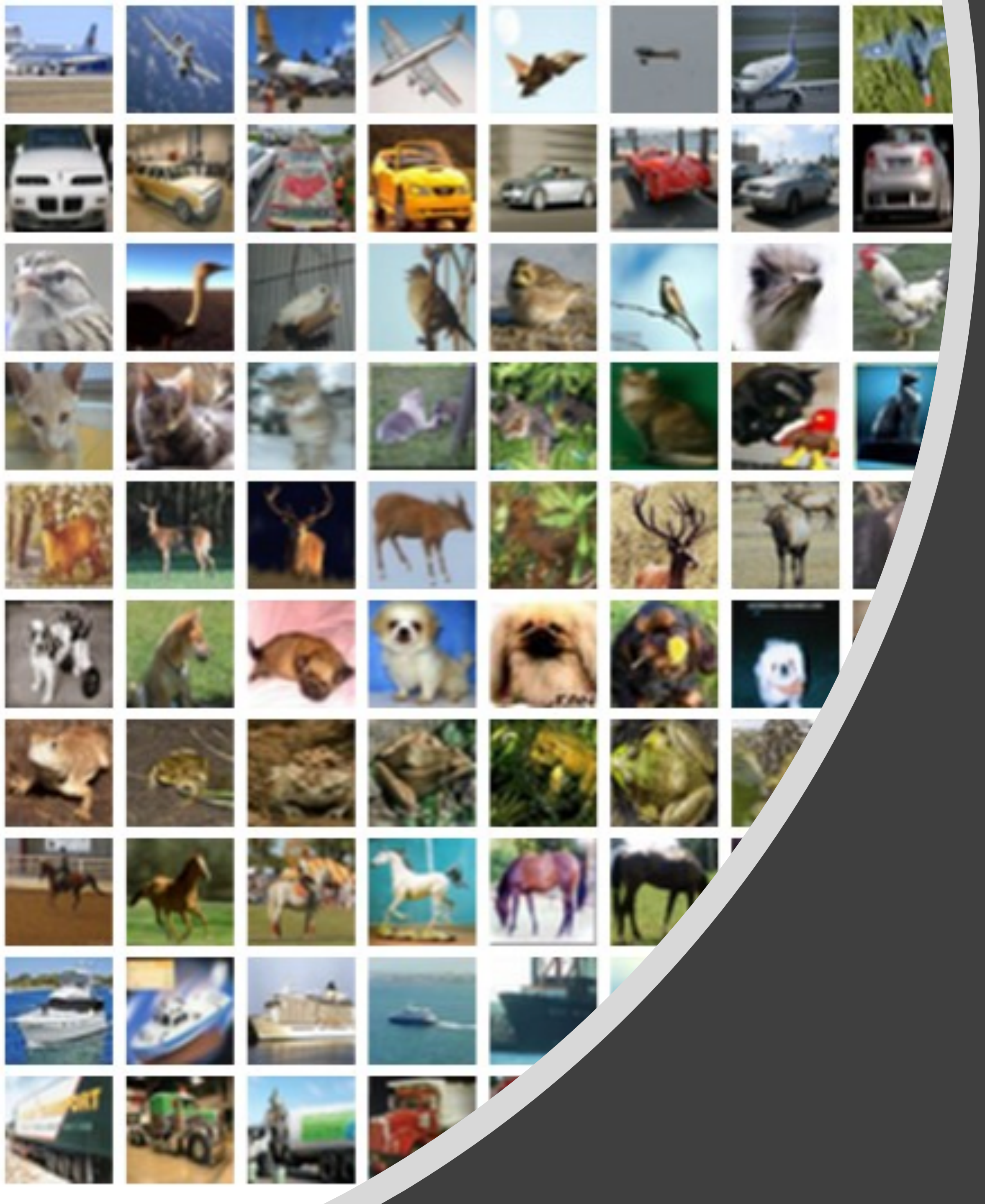
le



Overfitting and Avoidance

Plan for Class

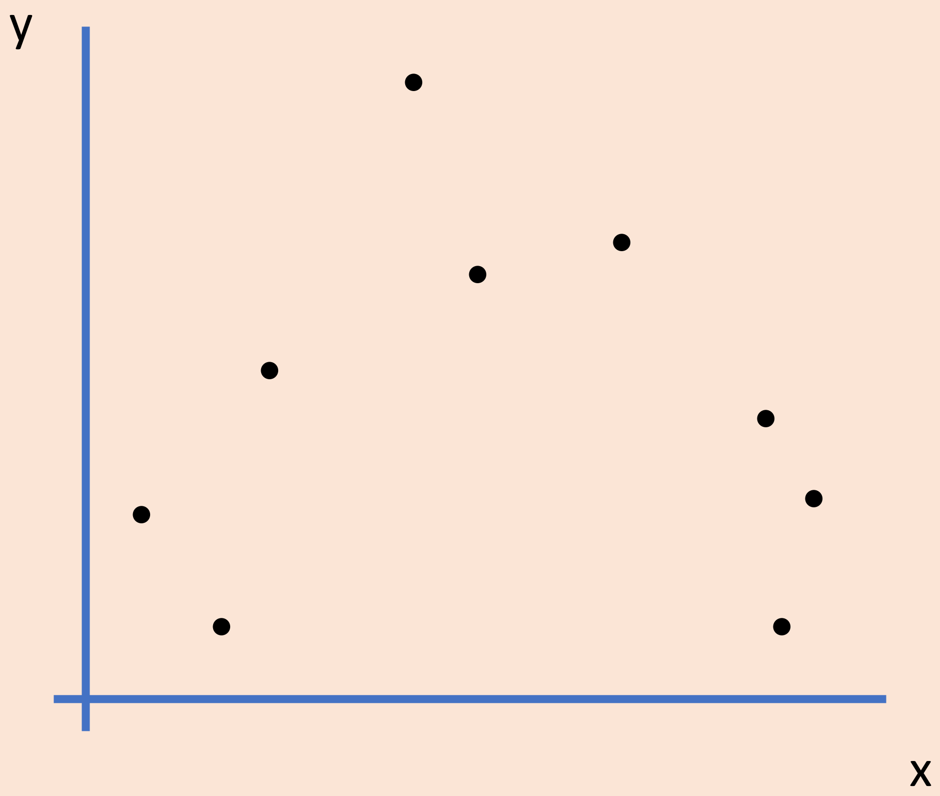
- ❑ Underfitting / Overfitting
- ❑ Overfitting and degradation of performance
- ❑ Training and Testing Data
  - ❑ Methods of splitting data
- ❑ Model Complexity
  - ❑ Overfitting in Trees
  - ❑ Overfitting in Neural Networks
  - ❑ Occam’s Razor
- ❑ Cross Validation
- ❑ Avoidance
  - ❑ Pruning
  - ❑ One in ten rule
  - ❑ Regularization

Fitting data

We begin with data that is generated through some process with observables y_t, x_t and an unknown function f

$$y_t = f(x_t) + w_t$$

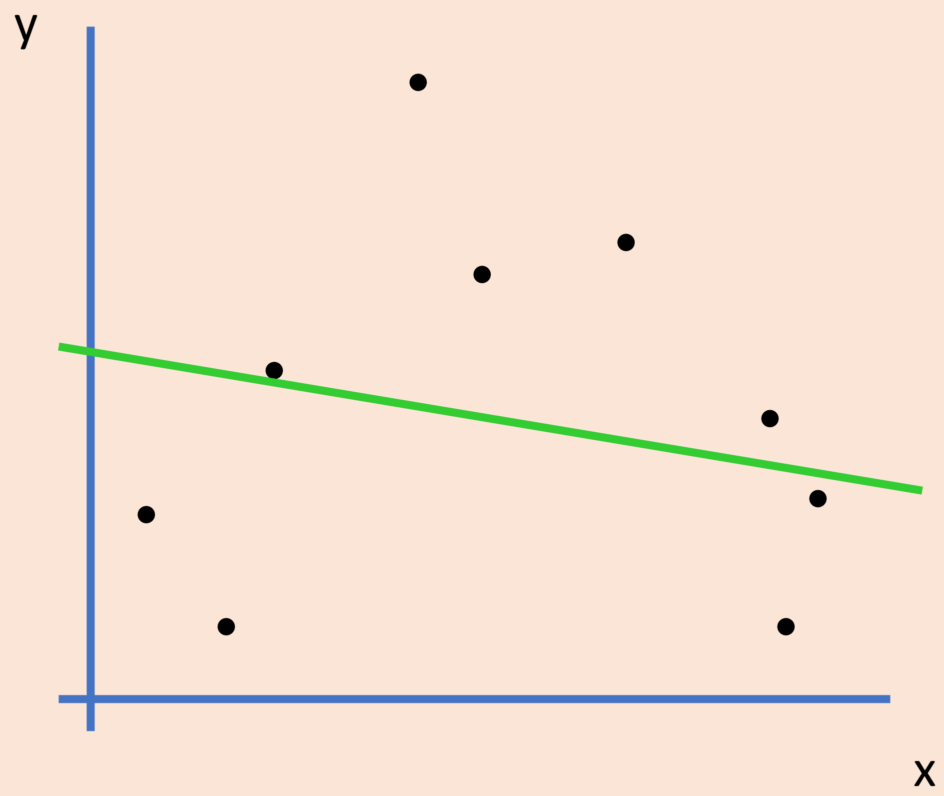
Can we learn f from this data?



Underfitting

Linear Regression

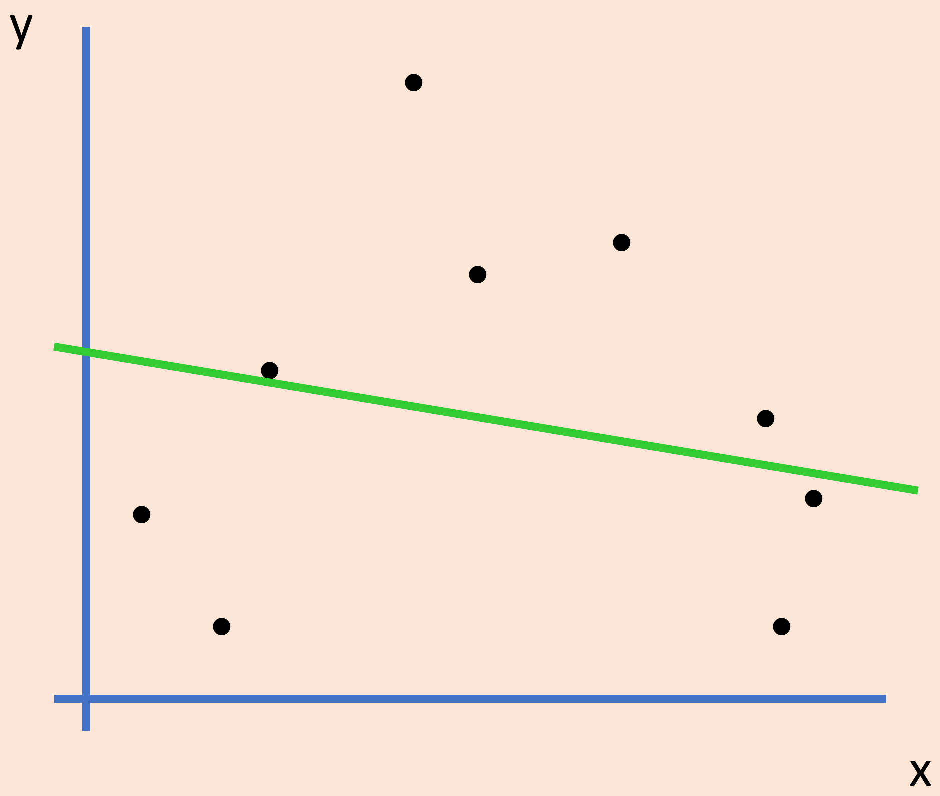
- A linear regression finds a linear function that minimizes a quadratic loss function
- If the underlying function is not linear then there will be a substantial difference between the expected value and the linear function this is referred to as ***underfitting*** or ***bias***



Underfitting

Underfitting happens when a model cannot capture the underlying trend of the data, in other words when the model does not fit the data.

.

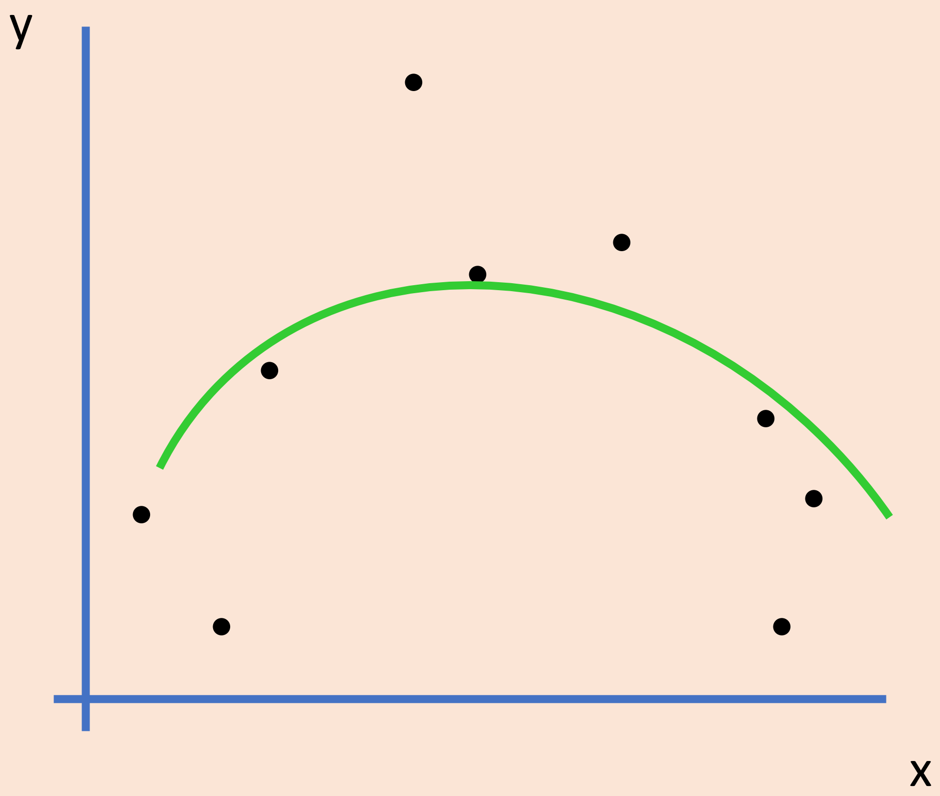


Fitting

Quadratic Regression

- A linear regression finds a quadratic function that minimizes a quadratic loss function

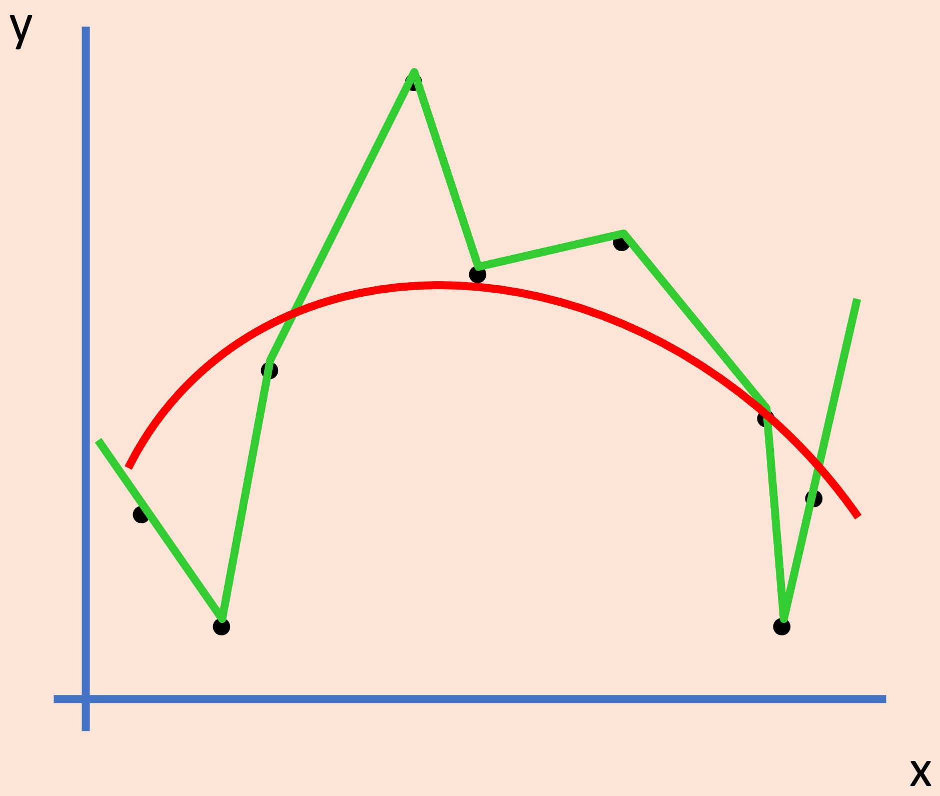
- As linear functions are also quadratic, this regression will necessarily reduce the bias. If the underlying is also quadratic we still expect to observe some loss due to noise



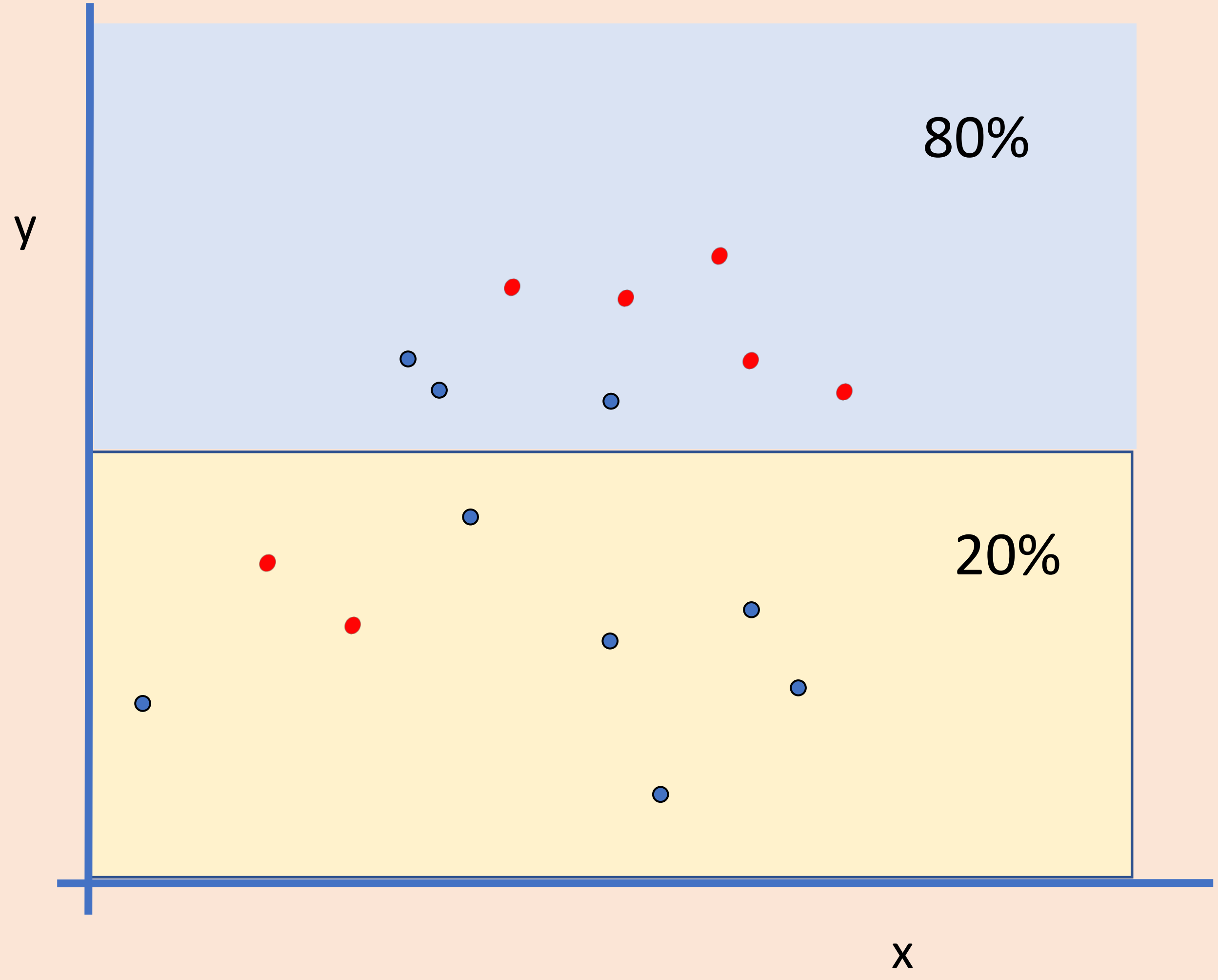
Overfitting

- There exists piecewise linear or high dimensional polynomials that pass through all points

- This function may not necessarily be a good forecast for points generated by the same process but not in the sample

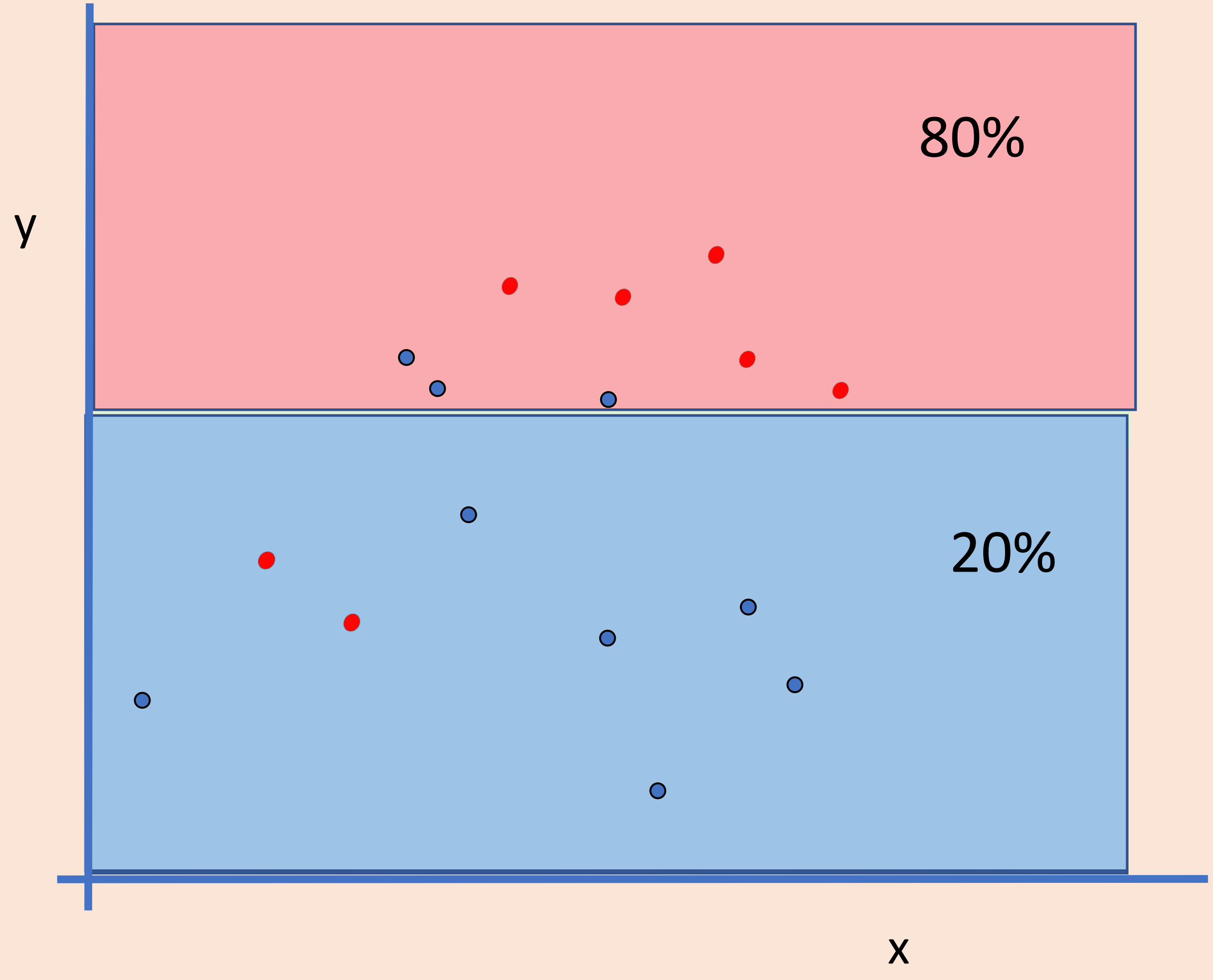


Overfitting a Decision Tree



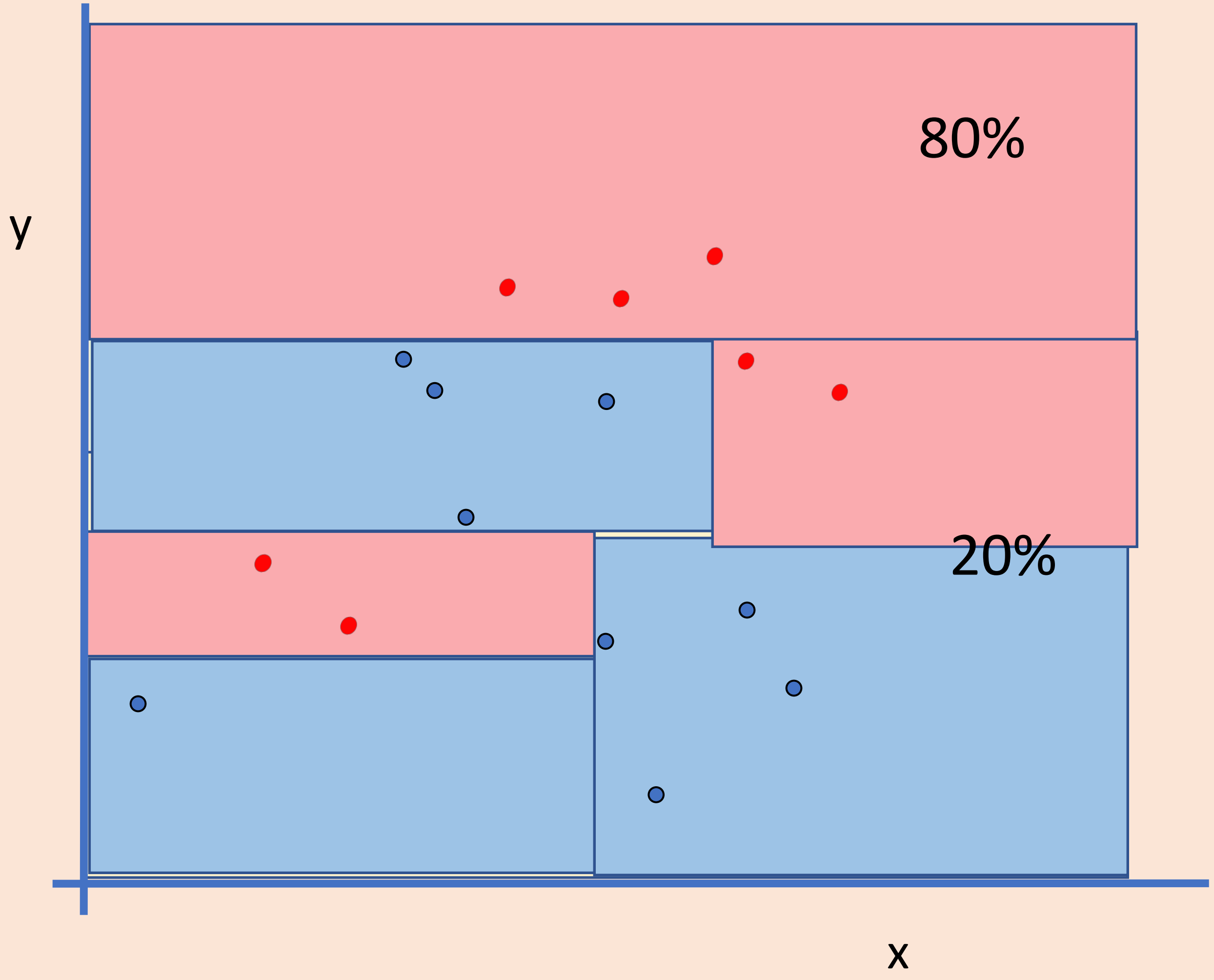
Overfitting a Decision Tree

- A three node decision tree will likely divide the space as follows
- In this case the likelihood of misclassification is 25%

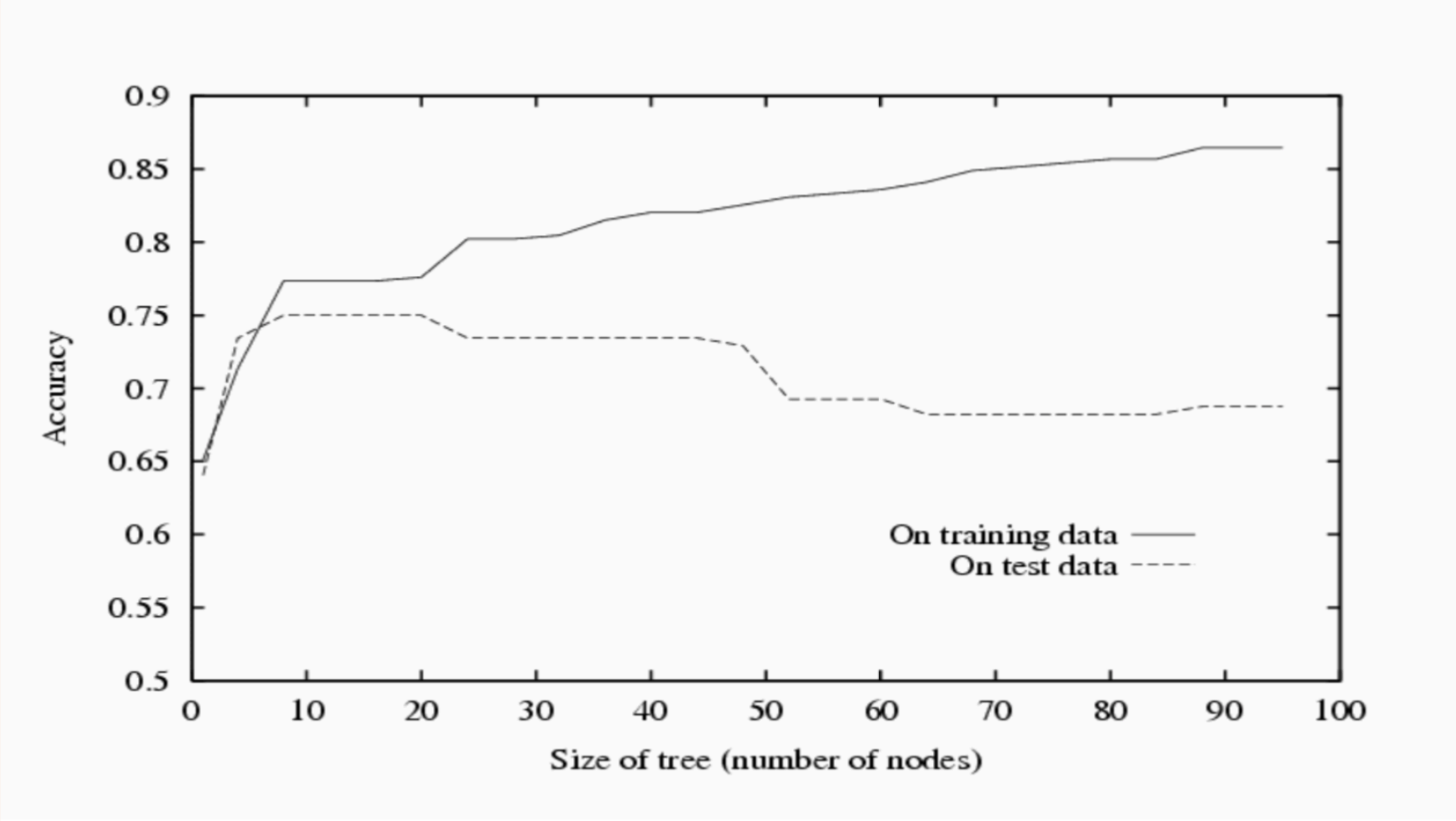


Overfitting a Decision Tree

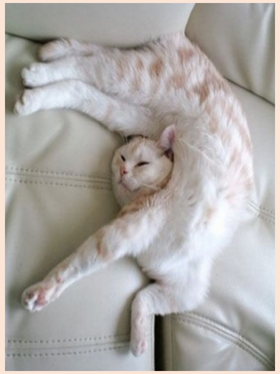
- A ten node decision tree divides the space as follows
- In case a point is drawn uniformly and classified by this decision tree the probability of misclassification is closer to 34%
- This shows that increasing the number of not=des decreases performance for new points

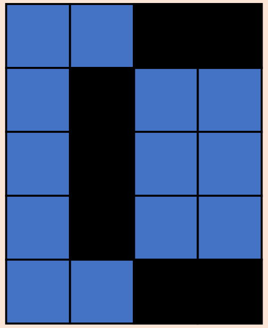


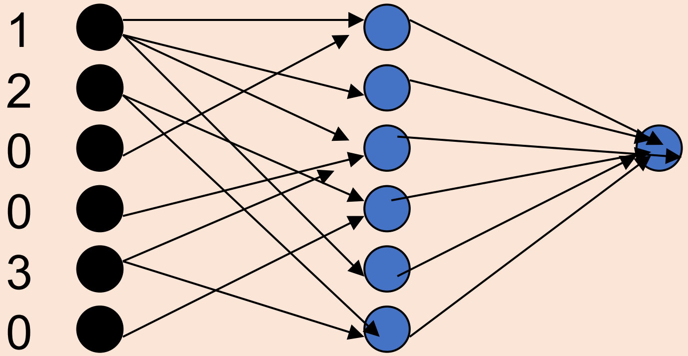
Overfitting a Decision Tree



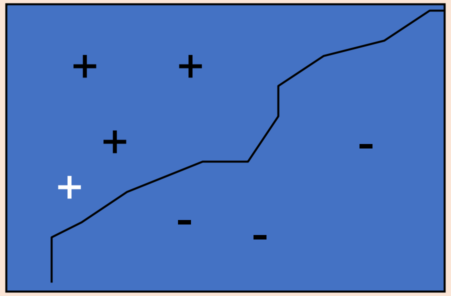
Neural Networks



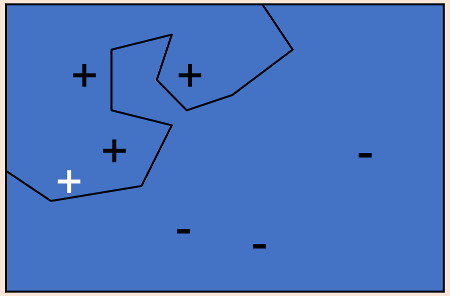




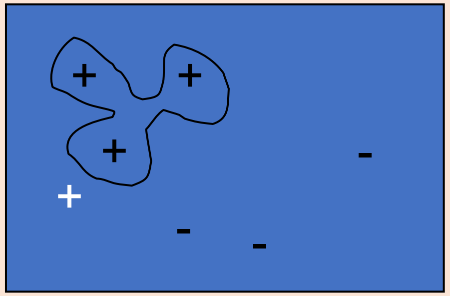
- Network trained on black samples;
- Curve is a visualisation of *decision space;* samples on one side are classified '+', and on the other as '-'.
- White samples are *out of sample / unseen,* aka part of the validation data



Good generalisation



Fairly poor generalisation



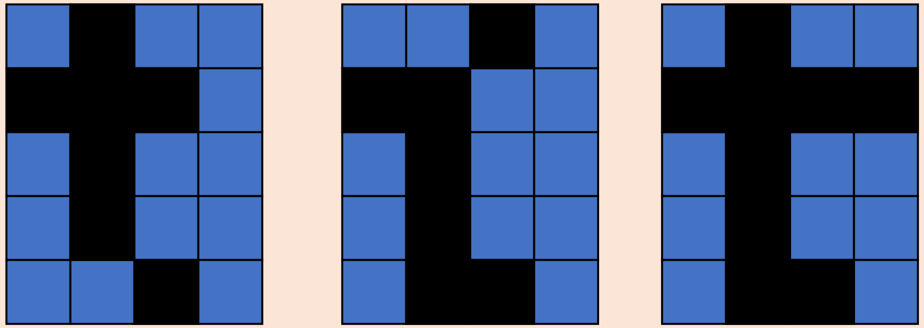
Overfit

Neural Networks

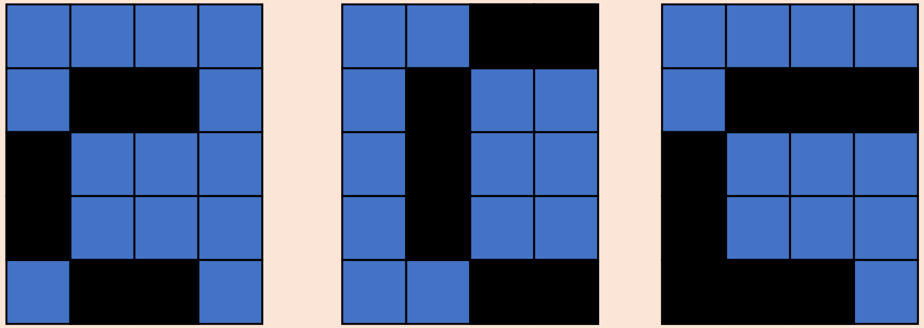
When we train an a classifier to tell the difference between handwritten t and c, using <u>only</u> these examples:

The classifier will learn easily. It will probably gives 100% correct prediction on these cases.

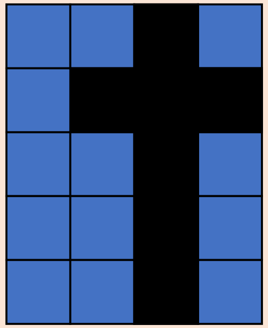
ts



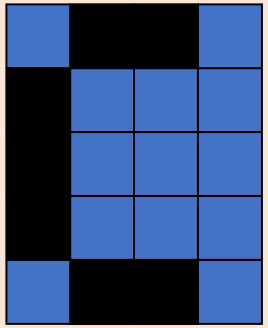
cs



Overfitting Neural Networks

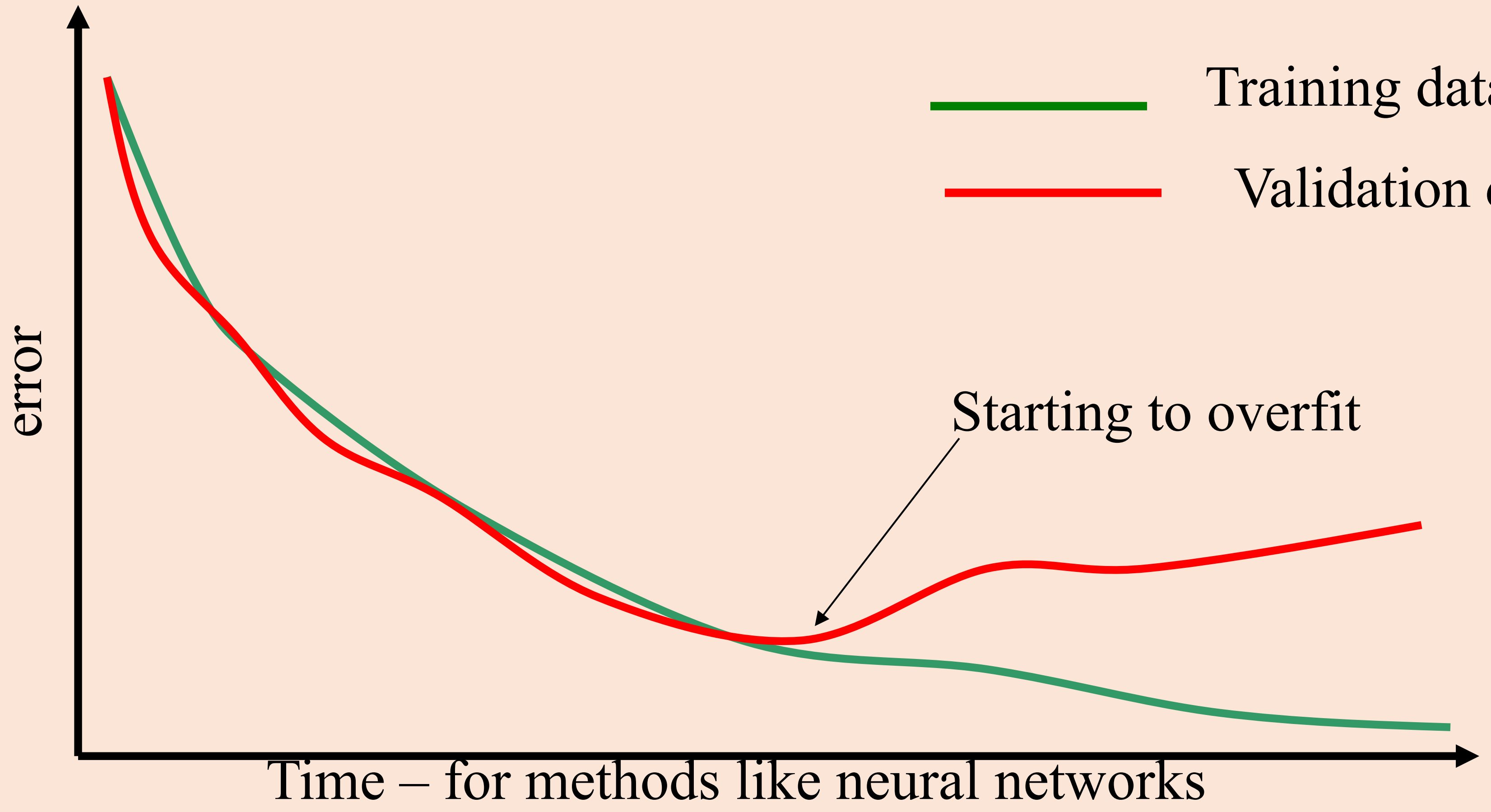


It will probably predict that this is a c



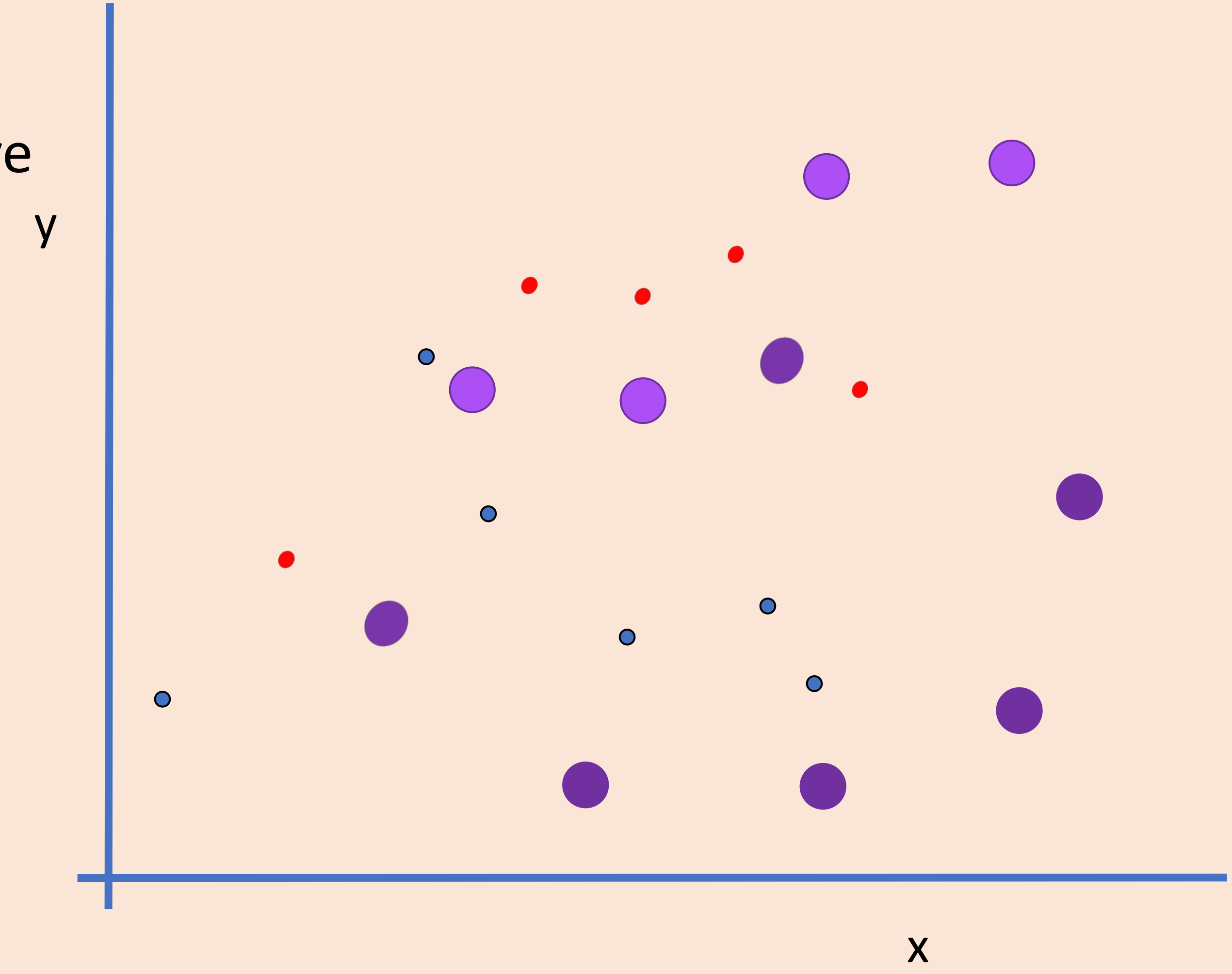
It will probably predict that this is a t

Neural Networks



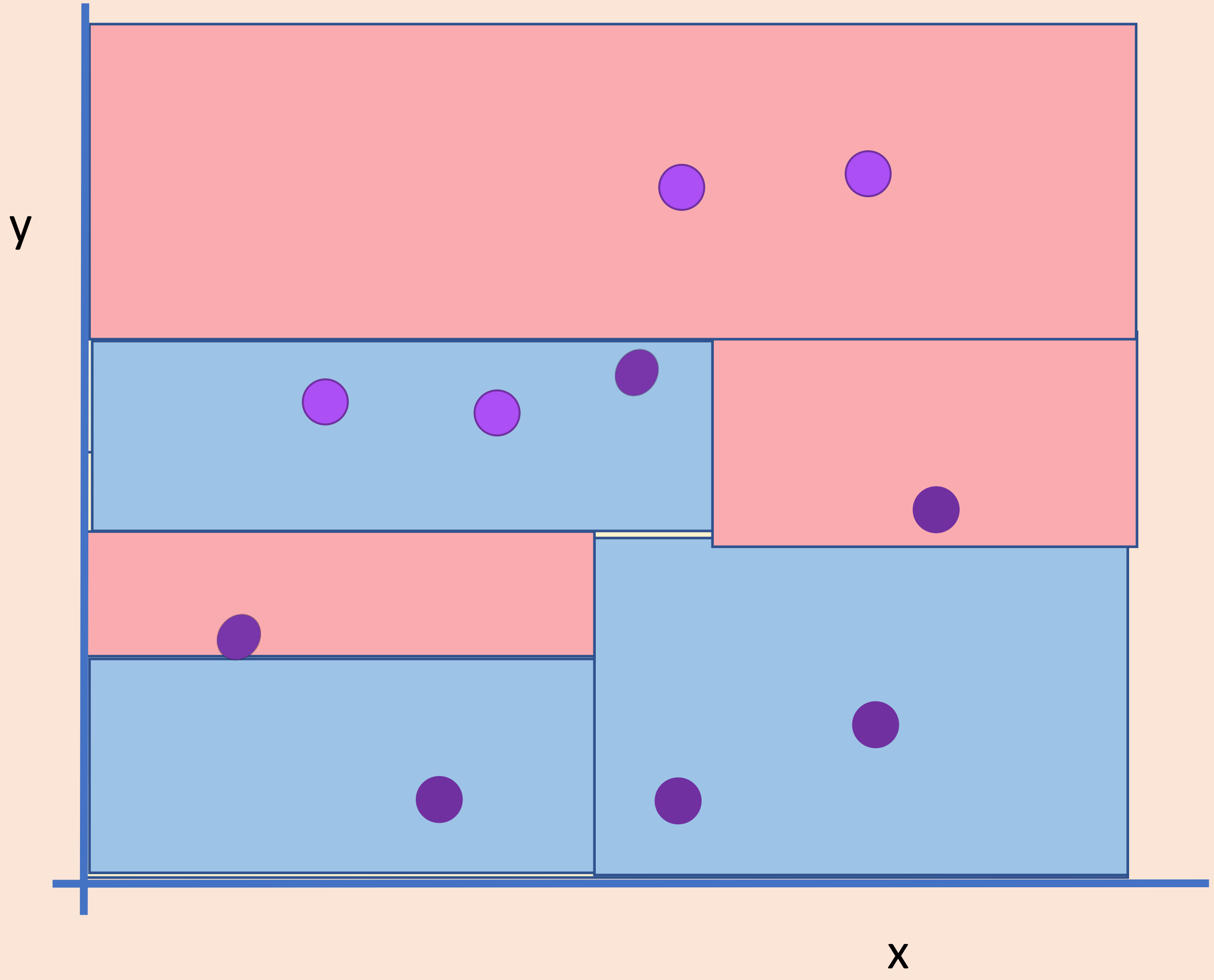
Model Validation

- To validate the model a certain fraction of the data is held in reserve
- The model is constructed using the remainder of the data
- The ***generalization*** of the model is then tested by the performance of the model on the reserved part of the data



Model Validation

- In the decision tree example, we see four instances misclassified giving us a 40% error rate



Generalization

- To validate the model a certain fraction of the data is held in reserve
- The model is constructed using the remainder of the data
- The ***generalization*** of the model is then tested by the performance of the model on the reserved part of the data
- The ***generalization error*** is a measure of how accurately the algorithm can forecast outcomes for data that has not been previously seen

Dividing Data

Care must be taken when dividing the data not to introduce a systematic difference between the test and train data

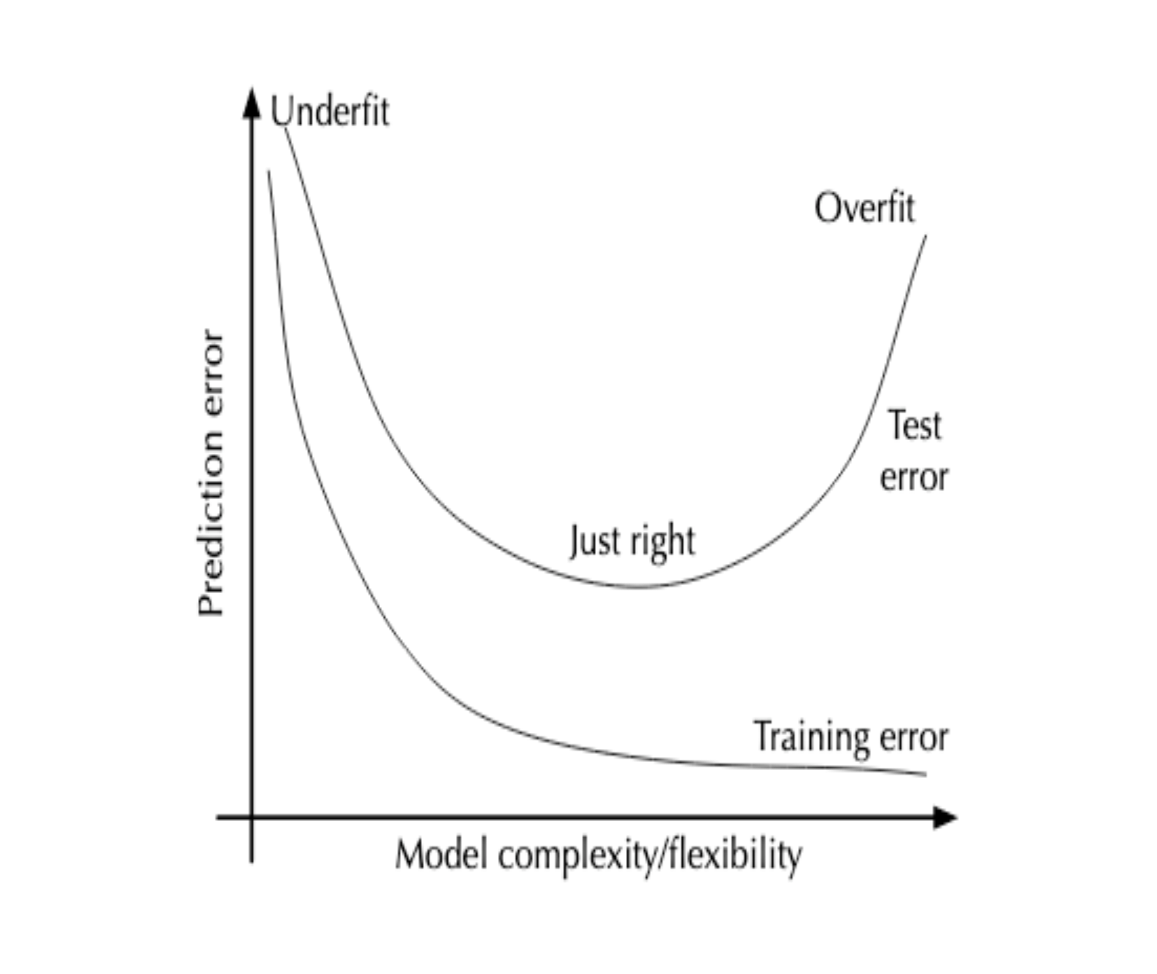
- Some cases random division would be sufficient

- For time series arbitrarily breaking the time continuity might introduce noise into the system
    - In such cases the usual convention is to divide days into two groups, or even between before and after some point in time

Complexity

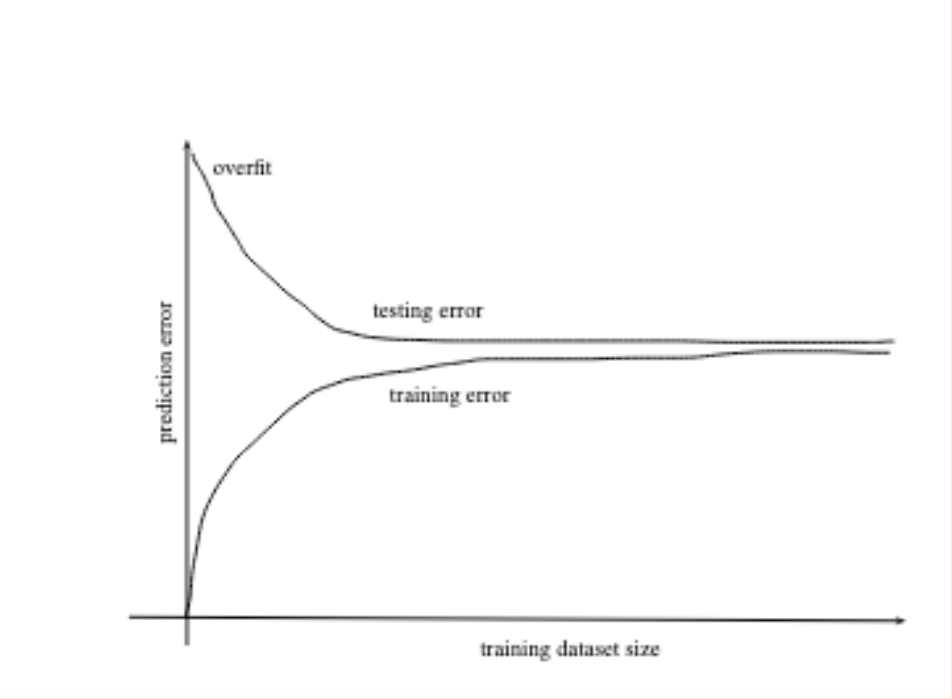
Complexity is the number of variables on which a model depends:

- For decision trees the key variables is the number of nodes and depth of the tree
- For linear and logistic regression it is the number of variables
- For neural networks if is the number of nodes in the net and the stopping time of the backpropagation algorithm



Learning Curve

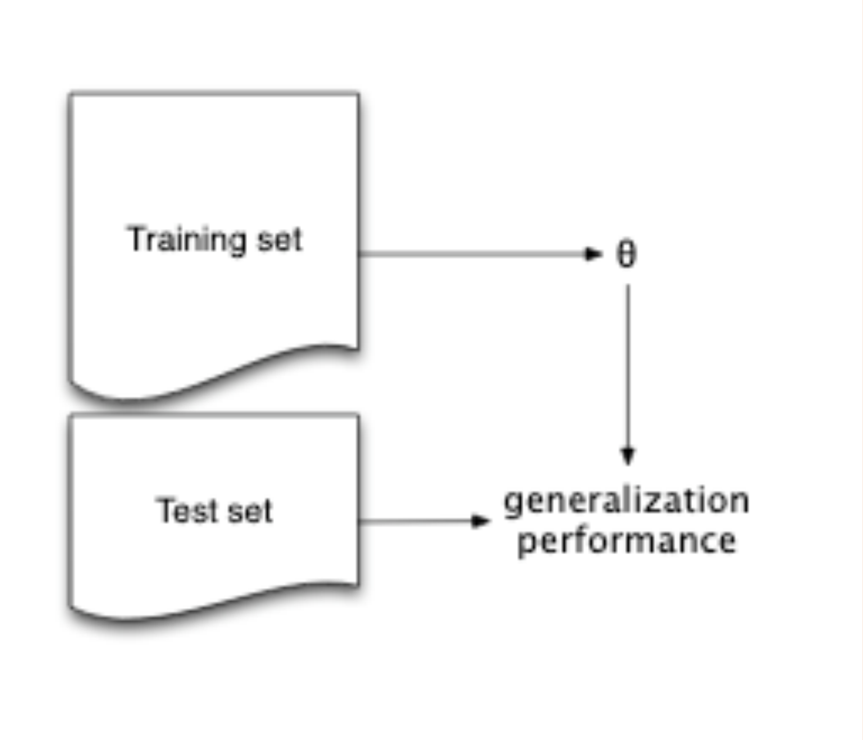
- The greater the training data, the greater the accuracy, but we must note there will always be some error due to noise in the system
- For a fixed level of complexity, when there is little training data, the model will overfit the training data leading to low error on the latter and greater error on out of sample data, aka generalization error



Optimizing Tuning Parameters

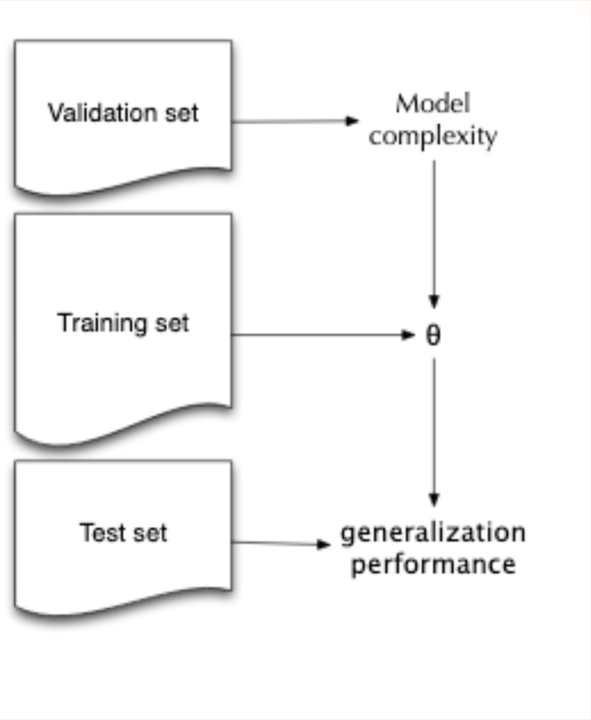
How can we optimize generalization ability, via optimizing choice of tuning parameters, model size, and learning parameters?

- Suppose we have split data into training/test set.
- Test set can be used to determine generalization ability, and used to choose best setting of tuning parameters/model size/learning parameters with best generalization.
- Once these tuning parameters are chosen, still important to determine generalization ability, but cannot use performance on test set to gauge this anymore!
- Idea: split data into 3 sets: training set, test set, andvalidation set.



Model Validation

- For each combination of tuning parameters
  - Train our model on the training set, fit parameters obtaining decision function f.
  - Evaluate average loss on a validation set.
- Pick parameters with best performance on validation set.
- Using these parameters train on both training and validation set to obtain the optimal parameters
- The new model is now a biased and can be overly optimistic
- Evaluate model with new parameters on test set, reporting generalization performance.



Cross Validation